PATENT CLAIMS

1. A preprocessor (2) for a predetermined document type definition (DTD), comprising

at least one predetermined interface for interchanging information with interfaces of application units (1, 6); and

a conversion means
- for converting application information from an application unit into calls to a markup language processor (3), with the calls satisfying the DTD, and
- for converting markup language information from the markup language processor (3) into return information for transmission to an application unit, with the return information being interpretable by the application unit.

2. The preprocessor (2) as claimed in claim 1, characterized in that
- the markup language processor is conforming with a predetermined API,
- the preprocessor has at least one interface for transmitting calls to the markup language processor and for receiving markup language information from the markup language processor, and
- the calls are API-conforming calls.

3. The preprocessor (2) as claimed in claim 1 or 2, characterized in that the markup language for the markup language document is XML, and the markup language processor (3) is an XML processor.

4. The preprocessor (2) as claimed in claim 3, characterized in that the API which is used is the Document Object Model (DOM).

5. The preprocessor (2) as claimed in one of claims 1 to 4, characterized in that said preprocessor is a substation configuration language (SCL) conforming preprocessor.

6. The preprocessor (2) as claimed in one of claims 1 to 5, characterized in that the application information and the calls include instructions.

7. The preprocessor (2) as claimed in one of claims 1 to 6, characterized in that the application information and the calls include structure information for building into a markup language file (5) which is processed by the markup language processor (3) and which is valid with respect to the DTD.

8. The preprocessor (2) as claimed in one of claims 1 to 7, characterized in that the return information includes structure information relating to a markup language file (5) which is processed by the markup language processor (3) and which is valid with respect to the DTD.

9. The preprocessor (2) as claimed in claim 8, characterized in that the structure information includes identifier information and/or content information.

10. The preprocessor (2) as claimed in one of claims 1 to 9, characterized in that the application information includes appliance configuration parameters for producing a markup language

document (5) for the configuration of at least one configurable appliance (6).

11. The preprocessor (2) as claimed in one of claims 1 to 10, characterized in that the return information includes appliance configuration parameters for an existing markup language document (5) for the configuration of at least one configurable appliance (6).

12. The preprocessor (2) as claimed in one of claims 1 to 11, characterized in that the conversion means comprises means for checking the syntax of the received information for conformity with the DTD.

13. The preprocessor (2) as claimed in one of claims 1 to 12, characterized in that the conversion means comprises means for checking the logical correctness and/or permissibility of structure information included in the information.

14. A system for processing valid markup language documents (5) which are conforming with a predetermined document type definition (DTD), comprising

an application unit (1, 6) for producing and/or reading in a set of application information items;

a preprocessor (2) as claimed in one of the claims 1 to 13; and

a markup language processor (3) for interchanging information with the preprocessor (2), for processing a markup language document (5) which is valid with respect to the DTD.

15. The system as claimed in claim 14, characterized in that the preprocessor (2) and the markup language processor (3) are combined to form a functional unit.

16. The system as claimed in claim 14, characterized in that the markup language processor (3) is a generic markup language processor (3) having an API-conforming interface (4) to the preprocessor (2).

17. The system as claimed in one of claims 14 to 16, characterized in that the application unit is a configuration program (1).

18. The system as claimed in one of claims 14 to 16, characterized in that the application unit is a configurable appliance (6).

19. The system as claimed in one of claims 14 to 18, characterized in that the application information items are appliance configuration parameters for configuring a configurable appliance (6).

20. The system as claimed in one of claims 14 to 19, characterized in that the predetermined DTD is the Substation Configuration Language (SCL).

21. A method for producing markup language documents (5) which conform with a predetermined document type definition (DTD), comprising the following steps:

- production of a set of application information items;

- production of an information representation, which is conforming with the predetermined DTD, from the application information; and

- production of a markup language document (5), which is valid with respect to the DTD, from the information representation.

22. The method as claimed in claim 21, characterized in that the application parameters are checked syntactically and/or semantically before the DTD-conformal information representation is produced from them.

23. The method as claimed in claim 21 or 22, characterized in that the application parameters are checked to ensure that the sense of their content is correct before the DTD-conformal information representation is produced from them.

24. The method as claimed in one of claims 21 to 23, characterized in that the process of producing the valid markup language document (5) comprises the following steps:

- production of calls from the DTD-conformal information representation, which are conforming with a predetermined API;

- transmission of the API-conformal calls to a markup language processor (3) via an interface (4) which is conforming with this API;

- execution of the API-conformal calls in order to process the valid markup language document (5).

25. The method as claimed in one of claims 21 to 24, characterized in that the application information items are appliance configuration parameters for at least one configurable appliance (6).

26. The method as claimed in one of claims 21 to 25, characterized in that the process of producing the API-conformal calls is carried out by means of a preprocessor (2) as claimed in one of claims 1 to 13.

27. A computer program product which can be loaded into an internal memory in a digital data processing means and which comprises computer program code means which execute the method as claimed in one of claims 21 to 26 when they are loaded and run in a data processing means.